

Магомедов А.М., Магомедов Т.А.

**АВТОМАТИЗАЦИЯ ПРОЦЕССА ПОДГОТОВКИ МАТЕРИАЛОВ
ТЕСТИРОВАНИЯ ПО ЯЗЫКАМ ПРОГРАММИРОВАНИЯ**

magomedtagir1@yandex.ru

Дагестанский Государственный Университет

г. Махачкала

Рассмотрен расширенный вариант процесса тестирования, включающий дополнительно к стандартным частям программного обеспечения генерацию тестовых элементов программным путем.

Структура тестового элемента. Представляется целесообразным сначала вкратце изложить принципиальную схему программного сопровождения тестирования. Для формулировки задачи, рассматриваемой в данном сообщении, достаточно (из соображений краткости) ограничиться упрощенной картиной. Пусть для тестирования отобраны нужные темы какой-либо учебной дисциплины, для каждой из этих тем подготовлен отдельный документ (далее – «тематический документ») из однотипных синтаксису и семантике и равноценных по трудности пунктов (далее – «тестовых элементов»). Программное обеспечение процесса тестирования условно разделим сначала на две части: программу генерации тестов и программу проверки результатов выполнения тестов студентами. Далее в сообщении предлагается дополнить эти две части начальной, «нулевой» частью.

Программа генерации тестов должна составить заданное количество уникальных тестовых документов, каждый из которых содержит требуемое количество тестовых элементов, выбранных по одному из каждого тематического документа в соответствии с тем или иным алгоритмом (или же случайным образом). Кроме собственно перечня пронумерованных тестовых элементов тестовый документ содержит *преамбулу* и *заключение*. Преамбула, помимо поясняющего текста в свободном формате, содержит несколько фиксированных полей, записи в которых служат целям идентификации (уникального тестового документа и студента). Из фиксированных полей заключения отметим основное поле – *таблицу*, в которую и предлагается записать ответы на тестовые элементы.

Программа проверки должна отсканировать и распознать цифровое содержимое фиксированных полей, затем – обеспечить сравнение списка выбранных студентом ответов со списком верных ответов (далее – «тестовых ключей»), соответствующих данному уникальному тестовому документу. Преемственность и обратная связь между программами проверки и генерации очевидны:

- программа проверки должна быть в состоянии определить «координаты» соответствующего списка верных ответов;

- эти координаты однозначным образом определяются значением фиксированного поля – идентификатора тестового документа;
- следовательно, программа генерации уже в процессе генерации тестового документа должна составить также и список верных ответов; в частности, при выборе очередного тестового элемента определить и соответствующий верный ответ для добавления к уже сформированной части списка верных ответов.

Для осуществления выбора тестового элемента необходимо выбрать определенный формат тематического документа. Предлагается включать верные ответы тестовых элементов в сами структуры, задающие тестовые элементы внутри тематических документов, путем встраивания элементов форматирования, помогающих программе извлекать их из соответствующих структур.

Для практической реализации схем, рассчитанных на массовое использование и предполагающих непременно массовое участие в подготовке тематических документов преподавателей – специалистов по «некомпьютерным» дисциплинам - важно учитывать особенности наиболее распространенных редакторов текста, а также стремиться к выбору элементов форматирования из числа наиболее простых (и в то же время – предоставляющих программе генерации теста достаточную информацию). Компромиссное решение выглядит следующим образом: 1) тестовый документ подготавливается редакторе текста и представляет собой набор тестовых элементов; 2) каждый тестовый пункт представлен одним абзацем (в терминологии предполагаемого редактора текста Microsoft Word); 3) после непечатаемого знака абзаца в тестовом пункте размещается верный ответ, который представлен либо одним (например, «минус»), либо несколькими символами, и заканчивается ограничительным символом, не включаемым в текст ответа (например, символом закрывающей скобки); 4) после ограничительного символа размещается та часть тестового элемента, которая программой генерации включается в тестовый документ; эта часть в качестве заключительного фрагмента включает определенное количество правдоподобных ответов, среди которых один или несколько верных. *Начальные фрагменты* всех ответов однотипны и обычно задаются цифрами; верный ответ, упомянутый в 3-м пункте настоящего перечисления, совпадает с одним из этих *фрагментов* или (в случае нескольких верных ответов) – с некоторым подмножеством множества этих *фрагментов*.

Пусть технология тестирования разработана и полностью отлажена в том смысле, что на базе созданных вручную тематических документов программа создает тестовые документы и тестовые ключи и осуществляет их распечатку, затем проводятся организационные мероприятия по тестированию студентов и сбору заполненных бумажных носителей; на заключительном этапе программа выполняет действия по их обработке. Создающееся представление о полноте и завершенности подобного технологического процесса ошибочно. Более того, многотрудная работа

только начинается. В следующем пункте рассмотрена проблема конструирования тестовых элементов программным путем.

Генерация тестовых элементов. На простом примере проверки знаний по языкам программирования рассмотрим вопрос автоматизации самого трудоемкого этапа подготовки материалов тестирования – создания множества тестовых элементов. Тем самым к двум частям программного обеспечения процесса тестирования будет добавлена начальная часть. Пусть требуется составить значительное количество однообразных тестовых элементов, представленных чередующимися последовательностями фрагментов вида

$$a[0] \ b[0] \ a[1] \ b[1] \ a[2] \ b[2] \ \dots \ a[n] \ b[n],$$

где $a[i]$ идентичны для всех однотипных тестовых элементов (*константные поля*), а поля $b[i]$ разных однотипных тестовых элементов (*параметрические поля*) меняются в диапазонах, определенных семантикой базового тестового множества. Например, структура тестового элемента в тематическом документе может иметь вид (параметрические поля выделены инверсным цветом):

-1) Укажите значение q после выполнения операторов:

q:='A=РОЗА_УПАЛА_НА_ЛАПУ_АЗОРА'; source:=q;

for k:=1 to length (q) do begin

n:=k; if q[k]='_' then break;

end;

q:=copy (q, 1, n-1);

for k:=n-5 downto 1 do begin

ch:=q[k]; q[k]:= q[n-4-k]; q[n-4-k]:=ch;

end;

q:=copy (source,n,3)+copy (q, 1, n-1);

Варианты ответов:

1) _УПА=Р 2) АЗОР=В 3) АЗБР=А 4) _УПАВР 5) _УП=Р

Сформулируем вкратце суть генерации тестовых элементов:

- программа случайным образом осуществляет перебор значений (в пределах допустимых диапазонов) параметрических полей за исключением полей

$$b[0], b[n-4], b[n-3], b[n-2], b[n-1], b[n], \quad (1)$$
- для каждого набора вычисляется значение q;
- вычисленное значение q присваивается последним пяти полям из (1), затем иницируется процесс случайного посимвольного искажения значений этих пяти полей, пока шесть значений: q, b[n-4], b[n-3], b[n-2], b[n-1], b[n] не будут попарно различны;
- затем генерируется случайное целое R из диапазона [1..5] и выполняется его запись в поле b[0], а запись верного ответа q - в соответствующее параметрическое поле b[n+5-R].

Программа последовательно подготавливает указанное в диалоговом режиме количество тестовых элементов, составляет тематический документ. В заключение посредством встроенных в среду программирования средств доступа к VBA выполняется преобразование тематического документа к требуемому формату: выполняется вставка уникального символа (скажем, ^) перед видимыми признаками начала тестового элемента, все знаки абзаца заменяются знаками мягкого переноса, после чего в позициях символов ^ восстанавливаются знаки абзаца.

Слепцова Л. Д. [Программирование на VBA в Microsoft Office 2007. Самоучитель](#). – М.: Диалектика, 2007.

Манжосов В.К., Новикова О.Д., Новиков Д. А.

**МОДЕЛИРОВАНИЕ НА ПЕРСОНАЛЬНОМ КОМПЬЮТЕРЕ
ЛАБОРАТОРНОЙ РАБОТЫ ПО СОПРОТИВЛЕНИЮ МАТЕРИАЛОВ
«ОПРЕДЕЛЕНИЕ ПЕРЕМЕЩЕНИЙ ПРИ ИЗГИБЕ БАЛКИ»**

nia@ulstu.ru

ГОУ ВПО Ульяновский государственный технический университет

г. Ульяновск

Представлены результаты математического моделирования процесса деформирования стержня при поперечном изгибе и определения прогиба в заданном сечении

Results of mathematical simulation of the process of deformation of a bar when cross bending and determination of deflection in the given section are presented.

При изучении дисциплины «Соппротивление материалов» предусмотрено проведение лабораторных работ, связанных с изучением процесса деформирования стержня при поперечном изгибе. Одна из работ предполагает проведение опытов по определению прогиба балки в заданном сечении.

Процесс освоения данной темы и подготовки к проведению лабораторной работы будет более эффективным, если осуществить компьютерное моделирование опытов.

При запуске программы на экране монитора (рис. 1) воспроизводится схема балки, установленной на шарнирных опорах, и панель управления процессом моделирования.